

Remarks

This reply is responsive to the Office communication mailed December 22, 2004. Unless otherwise indicated, page and paragraph references are to that communication.

Claims 1 and 4 have been amended to recite that program execution is transferred from an interpreter process "to a loop process" of a compiled code process, that the transfer at the transfer points is from the interpreter process "to said loop process" of the compiled code process, and that the transfer points are moved to the top "of said loop process" if they can be moved there without a problem occurring. While these limitations are believed to have been implicit in the claims as previously presented, they are now made explicit.

Claims 1-4 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,202,995 to O'Brien ("O'Brien") in view of U.S. Patent 6,513,156 to Bak et al. ("Bak") and Bacon et al., "Compiler Transformations for High-Performance Computing", ACM Computing Surveys, vol. 26, no. 4, Dec. 1994 ("Bacon") (page 5, ¶ 9).

Applicants will confine their current remarks to the O'Brien reference, since that reference underlies the outstanding rejection. As previously noted, O'Brien describes a method for removing "loop-invariant" conditional branches from instruction loops of a computer program. As shown in Figs. 9A-9B and described in the accompanying text, he accomplishes this by copying a loop (nodes B-F) containing an invariant conditional branch (node B) to form two parallel loops (Fig. 9B), moving the conditional branch instruction to a location (node A') before either loop, and having that moved instruction branch to one or the other of the two loops, depending on the value of the condition. This is also shown in the corresponding partial code listings in Figs. 6-8, where lines 100-200 in Fig. 6 correspond generally to node B in Fig. 9A.

While O'Brien is thus generally relevant to loop optimization, he does not teach moving transfer points to a loop process to the top of the loop, as claimed by applicants. Indeed, in the example shown in Fig. 9A, there are no transfer points at all to the loop B-F except for the original entry point B at the top of the loop. Furthermore, by its terms O'Brien's method operates only on

single-entry strongly connected regions (SESCRs)—i.e., loops having but a single entry point (col. 3, lines 7-22; col. 4, lines 10-17). Accordingly, since by hypothesis O'Brien has no transfer points to a loop other than at the top of the loop (i.e., the single entry point), he clearly does not suggest moving such transfer points to the top of a loop as claimed by applicants.

The Examiner argues (page 3, ¶ 5) that lines 150 and 200 of the code listing shown in Fig. 6 (corresponding generally to node B in Fig. 9A) constitute such transfer points. However, line 150 contains a conditional branch to line 250, within the original loop (lines 100-400), while line 200 contains an unconditional branch to line 300, also within the original loop. So if these lines are, broadly speaking, "transfer points", they are transfer points within the loop rather than transfer points into the loop (or entry points, using O'Brien's terminology) as claimed by applicants. Indeed, if lines 150 and 200 were such entry points, then, given the existing entry point at line 100, the loop would no longer be a singly-entry strongly connected region (SESCR) and would therefore be deemed ineligible for processing (col. 4, lines 10-17).

Clearly, then, O'Brien does not teach moving transfer points to a loop process to the top of the loop process as claimed by applicants. As a corollary, O'Brien also does not teach copying code from the top of a loop process to a point that post-dominates the top of the loop process and a transfer point to a location immediately preceding the loop process if the transfer point is located inside the loop process, as further claimed by applicants. Since, as noted above, there is no transfer point into O'Brien's loop process B-F except for the single entry point at node B, the condition for performing this claimed step of applicants' simply doesn't occur in O'Brien.

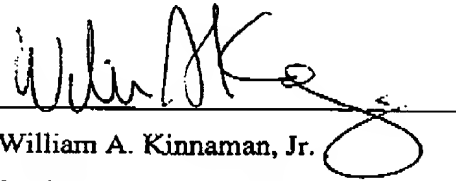
Given this basic failure of the primary reference of O'Brien to teach what it is asserted to teach—namely, (1) moving a transfer point to a loop process to the top of the loop process if it can be moved there without a problem occurring and (2) copying code from the top of the loop process to a point that post-dominates the top of the loop process and the transfer point to a location immediately preceding the loop process if the transfer point is located inside the loop process—the Examiner's combination of references must fail. Accordingly, claims 1-4, both as previously presented and as now amended, distinguish patentably over the art cited by the Examiner.

Conclusion

Entry of this amendment and reconsideration of the application as amended are respectfully requested. It is hoped that upon such consideration, the Examiner will hold all claims allowable and pass the case to issue at an early date. Such action is earnestly solicited.

Respectfully submitted,
TOSHIKI YASUE et al.

By



William A. Kinnaman, Jr.

Registration No. 27,650

Phone: (845) 433-1175

Fax: (845) 432-9601

WAK/wak